(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 711 480 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
11.06.1997 Bulletin 1997/24

(21) Application number: **93917657.4**

(22) Date of filing: **27.07.1993**

(51) Int Cl.[6]: **H04L 9/32**

(86) International application number:
PCT/EP93/01989

(87) International publication number:
WO 95/04416 (09.02.1995 Gazette 1995/07)

(54) **METHOD AND SYSTEM FOR AUTHENTICATED SECURE KEY DISTRIBUTION IN A COMMUNICATION SYSTEM**

VERFAHREN UND SYSTEM ZUR AUTHENTIFIZIERTEN SICHEREN SCHLÜSSELVERTEILUNG IN EINEM KOMMUNIKATIONSSYSTEM

PROCEDE ET SYSTEME DE DISTRIBUTION SECURISEE ET AUTHENTIFIEE DE CLES DANS UN SYSTEME DE COMMUNICATION

(84) Designated Contracting States:
DE FR GB IT

(43) Date of publication of application:
15.05.1996 Bulletin 1996/20

(73) Proprietor: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

(72) Inventors:
- **JANSON, Philippe**
  CH-8820 Wädenswil (CH)
- **TSUDIK, Gene**
  CH-8800 Thalwil (CH)

(74) Representative: **Barth, Carl Otto**
IBM Corporation
Säumerstrasse 4
8803 Rüschlikon (CH)

(56) References cited:
US-A- 4 649 233

- **COMPUTER SECURITY - ESORICS 92. SECOND EUROPEAN SYMPOSIUM ON RESEARCH IN COMPUTER SECURITY PROCEEDINGS 23-25 November 1992, Toulouse (FR) cited in the application**
- **IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATION vol. 11, no. 5 , June 1993 , NEW YORK US pages 679 - 693 XP000399664 R.BIRD ET AL. 'SYSTEMATIC DESIGN OF A FAMILY OF ATTACK-RESISTANT AUTHENTICATION PROTOCOLS'**
- **COMPUTER SECURITY - ESORICS 92. SECOND EUROPEAN SYMPOSIUM ON RESEARCH IN COMPUTER SECURITY PROCEEDINGS 23-25 November 1992, Toulouse (FR), pages 155-174; R. Molva et al.: "Krypto Knight Authentication and Key Distribution System"**

EP 0 711 480 B1

Description

TECHNICAL FIELD

The present invention relates in general to secure communication systems and in particular to secure cryptographic key distribution in a communication system. Still more particularly, this invention relates to secure authenticated key distribution in a communication system.

BACKGROUND OF THE INVENTION

Authentication protocols and key distribution protocols are known in the art.

A two-party authentication protocol (hereafter simply referred to as 2PAP) is described in R.Bird, I.Gopal, A.Herzberg, P.Janson, S.Kutten, R.Molva, M.Young: Systematic Design of Two-Party Authentication Protocols, Proceeding of Crypto '91, August 1991. In the described 2PAP, two users authenticate each others by transmitting challenges and using a shared secret key. This protocol has been shown to be secure against an important class of attacks known as interleaving attacks. Such attacks are based upon the adversary's ability to use either:

- legitimate flows obtained from past executions of the protocol or
- protocol flows elicited by the adversary from legitimate parties.

A family of key distribution protocols (hereinafter simply referred to as KDP) has been subsequently realized in an actual network security service, KryptoKnight, described in R.Molva, G.Tsudik, E. Van Herreweghen, S. Zatti: KryptoKnight Authentication and Key Distribution Service, Proceeding of ESORICS 92, October 1992, Toulouse, France.

In a two-party key distribution protocol (hereinafter simply referred to as 2PKDP) a user distributes to another user a new secret key, using an already previously secret shared key.

In a three-party key distribution protocol (hereinafter simply referred to as 3PKDP) and in a multi-party key distribution protocol (hereinafter simply referred to as MPKDP), as described in US-A-4,649,233, there is a user (often called "server") that shares a corresponding master user key with each of the other users; each master user key is known only to the server and the corresponding user. When a group of users (two or more) want to exchange secrets, since they do not share a common key, they have to rely on collaboration with the server. On request, the server generates and distributes a new group key for two (or more) selected users.

Key distribution is particularly difficult - because of time constraints - in communication system that provide for dynamic routing functions, enabling the continuation of routing, broadcasting, and multicasting (sending a message to a sub-set of users along a path or a set of paths) in the presence of changes. Such dynamic routing functions are per se known in the art, see for example Bertsekas-Gallager: Data Networks, Prentice-Hall, 1987, section 5.3.

SUMMARY OF THE INVENTION

There are some drawbacks with this prior art. The key distribution protocols known in the art have not beer shown sufficiently secure; at least not insofar as the 2PAP described by Bird et al., i.e., they are believed but have not been shown to resist any kind of attacks such as interleaving attacks or cut-and-splice attacks, for instance.

In addition, some do not guarantee integrity of the key being distributed, i.e. they do not guarantee that the key being distributed cannot be modified by an adversary. These limitations are particularly true of protocols with minimal-length messages for secure authenticated key distribution.

The above drawbacks of the prior art are overcome by the invention as claimed.

The present invention provides a method and system that have been mathematically demonstrated to be as secure as 2PAP, in particular against interleaving attacks, owing to a braided structure of the protocol messages. The basic two-party key distribution (and authentication) protocol can in turn be used as a basic building block for constructing more elaborate, e.g. three-party, multi-party or inter-domain, key distribution protocols. The braided structure, in addition, guarantees the integrity of the key being distributed. Besides, this authenticated key distribution protocol requires minimal length protocol messages, and it is as compact as possible in both the number and the size of messages exchanged.

All this can be achieved with minimal computational requirements and without the necessity of using encryption/decryption functions.

The foregoing and other objects, features, and advantages of the invention will be apparent from the following more particular description of a preferred embodiment of the invention, as illustrated in the accompanying drawing.

DESCRIPTION OF THE DRAWINGS AND NOTATIONS USED

The invention is described in detail below with reference to the following drawings:

FIG. 1 is a pictorial representation of a communication system which may be utilized for the implementation of the method and system of the present invention.

FIG. 2 depicts the 2PAP of the prior art.

FIG. 3 is a high level logical flow chart illustrating an example of the two-party authentication key distribution protocol according to the present invention.

FIG. 4 depicts the message flow of an example of 2PKDP according to the present invention.

FIG. 5 is a high level logical flow chart illustrating an example of the three-party authentication key distribution protocol according to the present invention.

FIG. 6 depicts the message flow of an example of 3PKDP according to the present invention.

FIG. 7 depicts the message flow of an example of MPKDP according to the present invention.

FIG. 8 depicts the message flow of an example of an optimized 3PKDP according to the present invention

The following terminology is adopted throughout the description:

A,B,C : full user names; all names are assumed to be distinct. These names are, for example, 64-bit values obtained either by directly coding the identity or by applying a hash function to reduce the original coding of the name into a 64-bit field.

S : full name of the authentication server; S is a user assumed to be universally trusted by all constituent users.

N_{ab} : all symbols beginning with N are random challenges or nonces, i.e. unpredictable, used-only-once random numbers; they can also be referred to as unit of freshness information. The first letter of the subscript refers to the user that originated the nonce while the second letter in the subscript identifies the challenged target user (e.g., N_{ab} is a nonce generated by A and sent to challenge B);

K_{ab} : all symbols beginning with K are keys. The subscript letters identify the users sharing that key (e.g., K_{ab} is the key shared between A and B);

$\oplus$: exclusive OR or XOR function.

E_k (X) : encryption of plaintext block X under key K. We use the term encryption to refer to a single-block transformation of cleartext input. This includes traditional encryption functions such as DES (described in National Bureau of Standards, Federal Information Processing Standards, National Bureau of Standards, Publication 46, 1977) as well as strong one-way functions (no one can invert) such as MD5, when the key is used as a suffix and/or prefix (described in R.Rivest: The MD5 Message Digest Algorithm, Internet DRAFT, July 1991). In the case of DES, both K and X are assumed to be not longer than the basic block size of the underlying encryption function, i.e. 64 bits. (Such assumptions are irrelevant for many one-way hash functions such as MD5. which accommodate any parameter size by definition.)

FUNC(x,y,z) : cryptographic function dependent upon arguments x,y,z; a function may depend on all or only some of its arguments. All functions generate results of a size equal to the basic block size of the underlying cryptographic algorithm, e.g. 64 bits for DES, 128 for MD5, etc.

A $\Rightarrow$ B X : this notation captures a single protocol flow; it denotes that A sent a message X to B.

GENERAL DESCRIPTION

With reference now to the figures and in particular with reference to Fig.1, a communication system (100) is represented. Users A(101), B(102), C(103) can reside in various locations which are computers, terminals, local area networks, connected by a set of links (104, 105, 106, 107, 108). One (or more) of these users may work as a server S (109) for secure key distribution. A common method is that a table of master user keys is kept in safe storage at the server (K_{as} is the master key shared by the server and a user whose identification is A; similarly K_{bs} is the key shared with B, and so on). Each user can also have his master user key stored in its safe storage device (encrypted file, special hardware, a smart card, etc.).

Fig. 2 shows the basic secure 2PAP, described by Bird et al. in the prior art, between two users (A and B) sharing a common key K_{ab}. The protocol execution begins at block 201 with the initiator party A transmitting its identity A and a nonce N_{ab} to the second user B. Upon receiving the message from A, the value of an authentication function AUTH (K_{ab},N_{ab},N_{ba},B), based upon the shared key K_{ab}, the nonce N_{ab}, a new nonce N_{ba} generated by B and the identity B of the message originator is computed by the second user B. Referring now to block 202, the value of this authentication function and the nonce N_{ba} are transmitted from B to A. Upon receiving the message from B, the first user A re-computes the value of the authentication function AUTH(K_{ab},N_{ab},N_{ba},B) and compares it with its counterpart in the message sent by B; a match results in authentication of B. The first user A then computes the value of another authentication function ACK(K_{ab},N_{ab},N_{ba}), based upon the shared key K_{ab}, the nonce N_{ab} and the nonce N_{ba}, in order to complete two-way authentication. Referring now to block 203, the value of the authentication function ACK(K_{ab},N_{ab},N_{ba}) is sent from A to B. Upon receiving the message from A, the user B re-computes the value of the authentication function ACK(K_{ab}, N_{ab},N_{ba}) and compares it with its counterpart in the message sent by A; a match results in authentication of A.

TWO-PARTY KEY DISTRIBUTION PROTOCOL (2PKDP)

Fig. 3 depicts an example of a secure two-party authenticated key distribution protocol between two users A and B according to the present invention. It is assumed that A and B share a common key K_{ab} prior to protocol execution.

The protocol execution begins at block 301 with the initiator party A transmitting its identity A and a nonce N_{ab} to the second user B, which can be considered as acting like a server to A. The second user B then generates a new key K_{ba} at block 302, which is held in a secure fashion and meant only to be used by A and B. The value $\hat{N}_{ba}$ of an authentication function AUTH(K_{ab},N_{ab},K_{ba},B), based upon at least the shared key K_{ab}, the nonce N_{ab}, the new key K_{ba} and the identity of B is computed by the second user B at block 303. The function AUTH(K_{ab},N_{ab},K_{ba},B) can be an authentication expression based on the shared key K_{ab}: one example of AUTH is (where all encryption is performed with key K_{ab}):

$$E(B \oplus (E(N_{ba} \oplus E(N_{ab}))).$$

The same user B then computes at block 304 the value of a function BRAID(K_{ab},$\hat{N}_{ba}$,K_{ba}), based upon at least the shared key K_{ab}, the value $\hat{N}_{ba}$ of the authentication function AUTH(K_{ab},N_{ab},K_{ba},B) and the new key K_{ba}. The function BRAID can be an encryption function or a masking expression MASK =$E_{Kb}(\hat{N}_{ba})$ computed by encryption of the value $\hat{N}_{ba}$ under the shared key K_{ab}, XOR-ed with the new key K_{ba}:

$$E_{Kab}(\hat{N}_{ba}) \oplus K_{ba}.$$

Referring now to block 305, the value of the authentication function AUTH(K_{ab},N_{ab},K_{ba},B) and the value of the function BRAID(K_{ab},$\hat{N}_{ba}$,K_{ba}) are transmitted from B to A. Upon receiving the message from B, at block 306 the first user A can extract the new key K_{ba} from the value of the function BRAID(K_{ab},$\hat{N}_{ba}$,K_{ba}), using the shared key K_{ab} and the value $\hat{N}_{ba}$. Assuming that the function BRAID is BRAID=$E_{Kab}(\hat{N}_{ba}) \oplus K_{ba}$, A computes the value of the masking expression MASK =$E_{Kab}(\hat{N}_{ba})$ and then extracts the new key K_{ba} by XOR-ing the value of the function BRAID=$E_{Kab}(\hat{N}_{ba}) \oplus K_{ba}$ with the value of the masking expression:

$$K_{ba}=E_{K_{ab}}(\hat{N}_{ba}) \oplus K_{ba} \oplus MASK = E_{K_{ab}}(\hat{N}_{ba}) \oplus K_{ba} \oplus E_{K_{ab}}(\hat{N}_{ba}).$$

If the user is indeed the user he claims to be by the identity sent to B in block 301, he has the key K_{ab} and can obtain the new key K_{ba}. If the user is impersonating the user A, he will not have the key K_{ab}, and will not be able to obtain the new key K_{ba}. Referring now to block 307, the first user A re-computes the value of the authentication function

AUTH(K_{ab},N_{ab},K_{ba},B) and compares it with its counterpart in the message sent by B; a match results in successful, authenticated (and secret) distribution of the new key K_{ba}. The process then passes to block 308, where the first user A computes the value of another authentication function ACK(K_{ak},N_{ab},K_{ba}), based upon at least the shared key K_{ab}, the nonce N_{ab} and the new key K_{ba}, in order to complete two-way authentication. The function ACK can be an authentication expression based on the shared key K_{ab}; one example of ACK is (where all encryption are performed with key K_{ab}):

$$E(K_{ba} \oplus E(N_{ab})).$$

Referring now to block 309, the value of the authentication function ACK(K_{ab},N_{ab},K_{ba}) is sent from A to B. The process ends at block 310, where the user B re-computes the value of the authentication function ACK(K_{ab},N_{ab},K_{ba}) and compares it with its counterpart in the message sent by A; a match results in authentication of A.

The distinctive feature of the braided structure of the BRAID function is important to achieve both authentication and, at the same time, key distribution. Moreover, the braided structure makes if possible to construct a minimal-length protocol message. This is because it is impossible to distribute an unpredictable random quantity (in secret) and simultaneously authenticate the quantity being distributed with a message of length less than two data units. The proposed two-party key distribution protocol is then extremely compact in both the number and the size of messages and this facilitates its application at any layer in the protocol hierarchy. Furthermore, as described in the following, the protocol is resistant to all interleaving attacks. All this is achieved with minimal computational requirements and without the necessity of using encryption/decryption functions (which are a stronger cryptographic tool, not always available).

The messages exchanged according to the proposed 2PKDP are illustrated in Fig. 4.

DISCUSSION OF SECURITY

For simplicity and clarity of exposition, we will refer to the example of 2PKDP shown in Fig. 4.

Authentication

The first step in demonstrating the security of the proposed 2PKDP is to show equivalence between it and 2PAP. In other words, we need to support the statement that the proposed protocol provides a means for two-party authentication with security equal to that of 2PAP. If we can proof this equivalence, then we argue that the proposed protocol inherits the basic properties of 2PAP, namely, resistance to interleaving attacks. Of course, protocol goals other than authentication cannot be inherited from 2PAP. In particular, the proposed protocol will be shown to meet the additional requirement of non-disclosure of the key being distributed.

We begin with some initial assumptions about key distribution protocols at large:

- A key is always generated by one party.
- A key being distributed must be a random, unpredictable quantity. If A and B engage in a 2PKDP and B generates a key, then neither A nor any other party can predict this key.
- Keys are never re-used.

These assumptions lead to an important observation that a freshly-generated key is very similar to a nonce; in fact, the only difference between a nonce N_{ba} and a new key K_{ba} is that N_{ba} is communicated in the clear while K_{ba} is kept secret.

We can observe now that the only difference between the two protocols is the so called 'nonce' field of the second message. In 2PAP (202), the nonce field is simply N_{ba}, while in 2PKDP (402) it is the more complex expression $E_{K_{ab}}(\hat{N}_{ba}) \oplus K_{ba}$. The purpose of this expression is to conceal the key, i.e. K_{ba}, which is used as a nonce (we consider the question of key confidentiality later). In order to demonstrate the equivalence of respective nonce fields, three issues must be addressed:

Is K_{ba} a nonce?

If K_{ba} is not a true nonce, then the equivalence does not hold. However, as noted earlier, nothing distinguishes a key from a nonce other than the fact that a key is kept secret. In particular, the rules and the procedures of generating keys and nonces are quite the same.

Is $\hat{N}_{ba}$ a nonce?

Since K_{ba} is a true nonce, $\hat{N}_{ba}$ is a result of a strong one-way function of (among other variables) K_{ba}, we conclude that $\hat{N}_{ba}$ is also a nonce.

Is there a one-to-one relationship between $E_{K_{ab}}(\hat{N}_{ba}) \oplus K_{ba}$ and K_{ba}?

Or, equivalently, does there exist a distinct value $\tilde{N}_{ba}$ such that: $E_{K_{ab}}(\hat{N}_{ba}) = E_{K_{ab}}(\tilde{N}_{ba})$. Under the assumption that $E_{K_{ab}}$ is a strong encryption function such as DES , this is clearly impossible since, otherwise, decryption would not be one-to-one. If, on the other hand, E is a strong one-way hash function (e.g. MD5) which produces a fixed-size digest of its input, different input values can be hashed into an identical digest. The issue is then the difficulty of computing $\tilde{N}_{ba}$. In MD5, for example, the probability of any given input hashing into a known digest (in our case, even the digest is unknown) is (impossible $\frac{1}{2^{128}}$ in practice).

Key-distribution

We now consider key distribution which is the added feature of the proposed 2PKDP. Strong authentication is not sufficient for achieving secure key distribution thus the results of the previous section are not applicable in this context.

The first issue is the definition of secure key distribution. For the purpose of key distribution, a protocol is considered secure if:

Non-disclosure:

The key being distributed cannot be obtained by the adversary (that is, without any assistance from either A or B).

Independence:

Knowledge of one key cannot be used to obtain other keys. Here we assume that the key being distributed will not be subsequently used as the encryption key in another 2PKDP. (The new key may, for example, be used for computing integrity checks on data messages exchanged between A and B).

Integrity:

The key being distributed cannot be modified by the adversary.

Non-disclosure

Assuming no collusion with a legitimate party, disclosure of a key is possible only if the adversary is somehow able to obtain the MASK value i.e., $E_{K_{ab}}(\hat{N}_{ba})$. However, this is impossible since each MASK is, in effect, a result of a one-way transformation of $\hat{N}_{ba}$. This implies that the adversary must either: 1) possess the encryption key, K_{ab}, or, 2) somehow be able to elicit the desired MASK value from one of the legitimate parties. The former is assumed to be a non-issue while the latter deserves a closer look.

The adversary can try to obtain the MASK value by interrogating B and pretending to be A. However, no matter what value the adversary sends to B, B always generates a fresh key in message (402). It is the freshness of the new key that makes it impossible for the adversary to obtain any useful information. In fact, it is quite irrelevant what the adversary tries to send to B since $\hat{N}_{ba}$ =AUTH (K_{ab},N_{ab},K_{ba},B) is strongly dependent on the freshly-generated K_{ba}.

Alternatively, the adversary can try to elicit information from A. This is significantly harder because A responds with message (403) only if message (402) is authentic. Therefore, the adversary must be able to generate expressions of the type in message (402) which is impossible since both cryptographic operations in message (402) are computed over A's random challenge, N_{ab}.

Independence

We now suppose that the adversary has somehow managed to discover a key K^i_{ba} from one of the runs of 2PKDP. Furthermore, he has done so without any explicit cooperation with either A or B. The most important additional piece of information that the adversary can thereby obtain is: $E_{K_{ab}}(\hat{N}^i_{ba}) = MASK^i$.

Now, since $MASK^i$ is the encryption of $\hat{N}^i_{ba}$ under the long-term key, K_{ab}, the adversary can discover a [plaintext, ciphertext] block-pair which he can later use for trying to break K_{ab}. This is, however, a danger commensurate with the presumed breach of security which resulted in the adversary's discovery of K^i_{ba}.

The remaining question is whether or not the adversary can use the newly-acquired information to attack the protocol by:

1. discovering a key K^i_{ba} ($i \neq j$) from any other protocol run (either past or future);

2. successfully distributing K^i_{ba} to A in another protocol run while posing as B.

As stated earlier, MASK is a result of a one-way function of $\hat{N}_{ba}$. Recall also that $\hat{N}_{ba}$ is, itself, a one-way function of N_{ab}, K_{ba} and B. This essentially precludes any possibility of type 1 attacks.

An attack of type 2 requires that the adversary be able to construct a message:

$$\hat{N}^j_{ba} = AUTH(K_{ab}, N^j_{ab}, K^i_{ba}, B),$$

$$E_{K_{ab}} (\hat{N}^j_{ba}) \oplus K^i_{ba}$$

where $i \neq J$.

This, in turn, is possible only if the adversary is able to produce a fresh value of $\hat{N}^j_{ba}$ which includes the computation of the AUTH expression over a fresh nonce N^j_{ab}. The adversary cannot obtain $\hat{N}^j_{ba}$ via oracle attacks since the party he chooses to interrogate will reply with a message implicitly re-freshed with its own nonce.

Integrity

The integrity of the key being distributed is not one of the foremost security features of a KDP. A pure key distribution protocol need not concern itself with the integrity of the key as long as the key cannot be modified to a particular value selected by the adversary. For example, some KDPs known in the art, as described in R.Bird at al. are vulnerable to key modification attacks since the key itself is not authenticated but, rather, only distributed. Nonetheless, this is not considered to be protocol a bug because the adversary cannot change the key to a known value; he can only offset a key by a known value.

On the other hand, attacks of this sort are not feasible with the present 2PKDP since key modification requires simultaneous modification of the authentication expression: $AUTH(K_{ab},N_{ab},K_{ba},B)$.

THREE-PARTY KEY DISTRIBUTION PROTOCOL (3PKDP)

Fig. 5 depicts a high level logical flow chart of a three-party authenticated key distribution protocol according to the present invention. The objective of a 3PKDP is to distribute a fresh group key to a group of users (two parties in this case). It is assumed that there exists a mutually-trusted third party typically referred to as the authentication server (S). It is assumed that all the users of the group share a corresponding secret master user key prior to protocol execution (K_{as} for A-S and K_{bs} for B-S). Each of said user key is know only to the server and the corresponding user. In the protocol of this example there is an initiator, denoted as A, which starts the communication. It is assumed that the group is determined in such a way that A knows who takes part in it. The protocol is started by an initiator A who first identifies the other parties (users) in the group.

A secure 3PKDP must satisfy much the same conditions as a secure 2PKDP with one added requirement:

- Neither party must be able to alter the group key being distributed.

More generally, we assume that one of the legitimate parties (A or B) may exhibit arbitrary 'byzantine' behavior (that it could impersonate another party that is registered with S). Of course, not all types of such threats can be addressed within the protocol. For example, unauthorized disclosure of a session key cannot be prevented. However, a secure 3PKDP must be defended against attacks by a legitimate party. In fact, a legitimate party exhibiting byzantine behavior can be viewed as a special case of an adversary (one armed with additional knowledge).

We obtain a secure 3PKDP by combining two instances (runs) of 2PKDP and one instance of 2PAP. Two users A and B each engage in a 2PKDP with the authentication server S. A and B are each assumed to share a user secret key with S, i.e., K_{as} and K_{bs} . After the distribution of the new key K_{ab}, A and B engage in a 2PAP in order to confirm each other's knowledge of K_{ab}.

Referring now to block 501, the protocol execution begins with a first user A transmitting its identity A, a nonce N_{as} and the identity of the users (only B in this case) of the group other than A to the server S. The server then generates a new group key K_{ab} at block 502, which is held in a secure fashion and meant only to be used by the users of the group (A and B in this example). The value $\hat{N}_{sa}$ of an authentication function $AUTH(K_{as},N_{as},K_{ab},B)$, based upon at least the user's key K_{as}, the user's nonce N_{as}, the group key K_{ab} and the identity of the users (B in this case) of the group

other than A is computed by the server at block 503. The server S then computes at block 504 the value of a function BRAID(K_{as},$\hat{N}_{sa}$,K_{ab}), based upon at least the user's key K_{as}, the value $\hat{N}_{sa}$ and the group key K_{ab}.

Referring now to block 505, the value of the authentication function AUTH(K_{as},N_{as},K_{ab},B) and the value of the function BRAID(K_{as},$\hat{N}_{sa}$,K_{ab}) are transmitted from S to A. Upon receiving the message from S, at block 506 the first user A can extract the group key K_{ab} from the value of the user's function BRAID(K_{as},$\hat{N}_{sa}$,K_{ab}), using the user's key K_{as} and the value $\hat{N}_{sa}$.

Referring now to block 507, the first user A re-computes the value of the user's authentication function AUTH(K_{as}, N_{as},K_{ab},B) and compares it with its counterpart in the message sent by S; a match results in successful, authenticated (and secret) distribution of the group key K_{ab}. The process then passes to block 508, where the first user computes the value of another authentication function ACK(K_{as},N_{as},K_{ab}), based upon at least the user's key K_{as}, the user's nonce N_{as} and the group key K_{ab}, in order to complete two-way authentication.

Referring now to block 509, the value of the authentication function ACK(K_{as},N_{as},K_{ab}) is sent from A to S. The first 2PKDP ends at block 510, where the server S re-computes the value of the authentication function ACK(K_{as},N_{as},K_{ab}) and compares it with its counterpart in the message sent by A; a match results in authentication of A.

Similar steps are now performed between the second user B and the server S. Referring to block 511, the second user B transmits its identity B, a nonce N_{bs} and the identity of the users (only A in this case) of the group other than B to the server S. The value $\hat{N}_{sb}$ of an authentication function AUTH(K_{bs},N_{bs},K_{ab},A), based upon at least the user's key K_{bs}, the user's nonce N_{bs}, the group key K_{ab} and the identity of the users (A in this case) of the group other than B is computed by the server at block 512. The server S then computes at block 513 the value of a function BRAID(K_{bs},$\hat{N}_{sb}$, K_{ab}), based upon at least the user's key K_{bs}, the value $\hat{N}_{sb}$ and the group key K_{ab}.

Referring now to block 514, the value of the authentication function AUTH(K_{bs},N_{bs},K_{ab},A) and the value of the function BRAID(K_{bs},$\hat{N}_{sb}$,K_{ab}) are transmitted from S to B. Upon receiving the message from S, at block 515 the second user B can extract the group key K_{ab} from the value of the user's function BRAID(K_{bs},$\hat{N}_{sb}$,K_{ab}), using the user's key K_{bs} and the value $\hat{N}_{sb}$.

Referring now to block 516, the second user B re-computes the value of the user's authentication function AUTH (K_{bs},N_{bs},K_{ab},A) and compares it with its counterpart in the message sent by S; a match results in successful, authenticated (and secret) distribution of the group key K_{ab}. The process then passes to block 517, where the second user computes the value of another authentication function ACK(K_{bs},N_{bs},K_{ab}), based upon at least the user's key K_{bs}, the user's nonce N_{bs} and the group key K_{ab}, in order to complete two-way authentication.

Referring now to block 518, the value of the authentication function ACK(K_{bs},N_{bs},K_{ab}) is sent from B to S. The second 2PKDP ends at block 519, where the server S re-computes the value of the authentication function ACK(K_{bs}, N_{bs},K_{ab}) and compares it with its counterpart in the message sent by B; a match results in authentication of B. The two users A and B then can authenticate each other by a 2PAP. Referring to block 520, the first user A transmits its identity A and a nonce N_{ab} to the second user B. Upon receiving the message from A, the value of the authentication function AUTH(K_{ab},N_{ab},N_{ba},B), based upon at least the group key K_{ab}, the nonce N_{ab}, a new nonce N_{ba} and the identity B of the message originator is computed by the second user B at block 521.

Referring now to block 522, the value of the authentication function AUTH(K_{ab},N_{ab},N_{ba},B) and the nonce N_{ba} are transmitted from B to A. Upon receiving the message from B, at block 523 the first user A re-computes the value of the authentication function AUTH(K_{ab},N_{ab},N_{ba},B) and compare it with its counterpart in the message sent by B; a match results in authentication of B. The process then passes to block 524, where the first user computes the value of the authentication function ACK(K_{ab},N_{ab},N_{ba}), based upon at least the group key K_{ab}, the nonce N_{ab} and the nonce N_{ba}, in order to complete two-way authentication.

Referring now to block 525, the value of the authentication function ACK(K_{ab},N_{ab},N_{ba}) is sent from A to B. The process ends at block 526, where the second user B re-computes the value of the authentication function ACK(K_{ab}, N_{ab},N_{ba}) and compare it with its counterpart in the message sent by A; a match results in authentication of A.

The messages exchanged according to the proposed 3PKDP are illustrated in Fig. 6.

We note that the server can also play the role of a user (in addition to server). The server can use the protocol to share a key with a group which contains the server itself (as a member of the group), in which case the server can play both the role of the protocol server and the role of the user, omitting sending the group key and authentication verification to itself (which is not needed), but only to a sub-group of users consisting of the users of the group other than the server S. In particular, this can be used to refresh the keys shared by the server and a user in a reliable way that makes sure that only the server and the actual user can change the key they share. We note that if the group consists only of two users (e.g., A and S), the protocol becomes equal to the described 2PKDP.

DISCUSSION OF SECURITY

For simplicity and clarity of exposition, we will refer to the example of 3PKDP shown in Fig. 6.

Authentication

As with 2PKDP, the first issue we must address is the strength of 3PKDP as an authentication protocol. In other words, does the protocol achieve pairwise authentication of (A,S), (B,S) and, finally, (A,B)?.

Starting from the end of the protocol, it can be easily seen that, assuming secure distribution of K_{ab}, the authentication protocol between A and B is as secure as the underlying 2PAP (in fact, it is 2PAP.)

Viewed in isolation, 2PKDP between A and S is secure and so is 2PKDP between B and S. However, one important difference between 3PKDP and two unrelated runs of 2PKDP is that the nonce generated by S is the same in both runs of 2PKDP (i.e., K_{ab}). This is not a problem as long as it can be shown that the respective AUTH expressions remain uniquely 'tied' to the particular run of 3PKDP. We address this issue by explicitly including the name of the peer party in the calculation of each AUTH expression, i.e., B is included in AUTH with K_{as} and A in AUTH with K_{bs}. While this argument is not particularly crisp, we note that one simple method for complete conformance with 2PKDP is to use $K_{ab} \oplus A$ instead of K_{ab} in AUTH with K_{as} and, similarly, $K_{ab} \oplus B$ instead of K_{ab} in AUTH with K_{bs}.

Key distribution

The non-disclosure and independence properties in 3PKDP are inherited directly from 2PKDP. The integrity of the new key is a different matter. The side-effect of the same key being distributed to both parties, is that an 'insider' adversary, say B, can obtain the expression used to mask the key in the message from S to A.

$$MASK_a = E_{Kas}(\hat{N}_{sa}) = E_{K_{as}}(AUTH(K_{as},N_{as},K_{ba},B)).$$

At the first glance, it appears that knowledge of this expression allows B to modify K_{ab} to any selected $\bar{K}_{ab}$ with impunity. Nonetheless, the AUTH with K_{as} expression (enclosed in the same message) precludes this attack since B cannot alter $AUTH(K_{as},N_{as},K_{ab},B)$ to be $AUTH(K_{as},N_{as},\bar{K}_{ab},B)$

On a related note, one advantage that B does gain as a result of 3PKDP is the knowledge of the cleartext-ciphertext pair of the form: $\hat{N}_{sa}$, $E_{K_{as}}(\hat{N}_{sa})$. This poses a threat insofar as B is able to discover K_{as} by means of a brute-force attack. However, the amount of work required on the average remains considerable, certainly commensurate with what would be required to crack an AUTH expression in the first place.

MULTI-PARTY KEY DISTRIBUTION PROTOCOL

Fig. 7 depicts the list of the messages exchanged in an example of a secure multi-party authenticated key distribution protocol from a server S to a group of selected users (A, B and C in the example) according to the present invention. In the protocol of this example there is an initiator, denoted as A, which starts the communication. It is assumed that the group is determined in such a way that A knows who is part of it. It is also assumed that all users of the group share a corresponding secret user key prior to protocol execution (K_{as} for A-S, K_{bs} for B-S and K_{cs} for C-S). Each of said user keys is known only to the server and a corresponding one of the users. The protocol is started by the initiator A who first identifies the other parties (users) in the group.

The actual process flow implementing the above protocol messages can be plainly derived from Fig. 7 if the detailed discussion of the actual process flow of the previous examples is taken into account.

The Multi-Party Key Distribution Protocol (MPKDP) is obtained by combining three runs of 2PKDP: A-S, B-S, and C-S with three runs of 2PAP: A-B, A-C, B-C. The goal of the MPKDP is to distribute to all parties the new key K_{abc}. After receiving K_{abc}, A-B, A-C, and B-C engage in a 2PAP to confir each other's knowledge of K_{abc}.

Referring to blocks 701, 704 and 707, each of A, B and C, sends to server S its identity, its own nonce (N_{as}, N_{bs}, N_{cs}), and the identities of other group members: (B,C) for A, (A,C) for B and (A,B) for C.

Referring to blocks 702, 705 and 708, server S generates a new key K_{abc} which is held in a secure fashion and is meant only to be used by the members of the group (A, B, and C in this example.) The values of $\hat{N}_{sa}$, $\hat{N}_{sb}$, and $\hat{N}_{sc}$ are: $AUTH(K_{as},N_{as},K_{abc},B,C)$, $AUTH(K_{bs},N_{bs},K_{abc},A,C)$ and $AUTH(K_{cs},N_{cs},K_{abc},A,B)$, respectively. Each is based upon at least the user's key (K_{as}, K_{bs} or K_{cs}), the user's nonce (N_{as}, N_{bs} or N_{cs}), the group key K_{abc}, and the identities of other group members ([B,C], [A,C] or [A,B]).

Server S also computes the value of the functions: $BRAID(K_{as},\hat{N}_{sa},K_{abc})$, $BRAID(K_{bs},\hat{N}_{sb},K_{abc})$, and $BRAID(K_{cs},\hat{N}_{sc},K_{abc})$. Each of these functions is computed as in the description of 3PKDP.

Finally, server S sends out

to A: $AUTH(K_{as},N_{as},K_{abc},B,C)$ and $BRAID(K_{as},\hat{N}_{sa},K_{abc})$;

to B: AUTH(K_{bs},N_{bs},K_{abc},A,C) and BRAID(K_{bs},$\hat{N}_{sb}$,K_{abc});
to C: AUTH(K_{cs},N_{cs},K_{abc},A,B) and BRAID(K_{cs},$\hat{N}_{sc}$,K_{abc}).

Each of A, B, and C verifies its respective message and extracts the new key K_{abc}.

Referring to blocks 702, 705 and 709, each user A, B, and C computes and sends to server S the value of ACK (K_{as},N_{as},K_{abc}), ACK(K_{bs},N_{bs},K_{abc}) and ACK(K_{cs},N_{cs},K_{abc}) whereof each is based upon at least the user's key (K_{as}, K_{bas}, K_{cs}), the user's nonce (N_{as}, N_{bs} or N_{cs}), and the group key K_{abc}. Upon receipt of all three messages, S verifies each one using the corresponding values. This completes the authenticated key distribution.

The remainder of the protocols (blocks 710 through 718) is devoted to three runs of 2PAP to confirm to each of the three group members that the other members hold the new key K_{abc}. Each of the 2PAP runs A-B, A-C, and B-C is identical to that in Fig.2.

OPTIMIZATION

The protocol shown in Fig. 6 (similar considerations can be made about the protocol of Fig. 7) was obtained by a straightforward combination of 2PKDPs and 2PAPs. It requires a total of 9 messages and 3 bi-directional communication flows (A-B, A-S, B-S). In order to reduce the number of messages and optimize the protocol some of the messages can be interleaved. Moreover, it is not necessary for both A and B to communicate with S directly. A more realistic scenario would be such that all communication with S is either through A or B.

Referring now to Fig. 8, there is depicted a compacted version of a 3PKDP. In this example A starts the execution, then passes its information to B which passes A's information to S together with B's information. In a counter-move, the server S sends all the information to B, that distributes its piece of information to A. In order to make the protocol insensitive to the order of communication inside the group of users communicating with each other, tags are used so that a particular user knows which particular piece of information it is to use in a stream of data.

Referring now to block 801, the protocol execution begins with an initiating user A transmitting his identity, his nonce for S (N_{as}), his nonce for B (N_{ab}) and the name of the other user (B) to that user B.

At block 802, upon receipt, user B transmits to S his own identity, identity of the other use (A), his nonce for S (N_{bs}) and A's nonce for S (N_{as}).

Referring now to block 803, server S generates a new key K_{ab} which is held in a secure fashion and meant only to be used by A-and B. The value $\hat{N}_{sa}$ of an authentication function AUTH(K_{as},N_{as},K_{ab},B) based upon at least the user's key K_{as}, the user's nonce N_{as}, the new key K_{ab}, and the identity of the other user B is computed. Server S then computes the value of a function BRAID(K_{as},$\hat{N}_{sa}$,K_{ab}), based upon at least the user's key K_{as}, the value $\hat{N}_{sa}$, and the new key K_{ab}.

Referring still to block 803, server S also computes the value $\hat{N}_{sb}$ of an authentication function AUTH(K_{bs},N_{bs},K_{ab}, A) based upon at least the user's key K_{bs}, the user's nonce N_{bs}, the new key K_{ab} , and the identity of the other user A. Server S then computes the value of a function BRAID(K_{bs},$\hat{N}_{sb}$,K_{ab}), based upon at least the user's key K_{bs}, the value $\hat{N}_{sb}$, and the new key K_{ab}.

Finally, server S transmits AUTH(K_{as},$\hat{N}_{as}$,K_{ab},B), BRAID(K_{as},N_{sa},K_{ab}), AUTH(K_{bs}, N_{bs} ,K_{ab},A) and BRAID(K_{bs},$\hat{N}_{sb}$, K_{ab}) to B.

Referring now to block 804, user B receives the above message and extracts the new key K_{ab} from the value of the user's function BRAID(K_{bs},$\hat{N}_{sb}$,K_{ab}) using the key K_{bs} and the value AUTH(K_{bs},N_{bs},K_{ab},A). Then, B authenticates the new key K_{ab} by verifying the authentication function AUTH(K_{bs},N_{bs},K_{ab},A) based upon at least the user's key K_{bs}, the value N_{bs}, the new key K_{ab}, and the identity of the other user A.

Then, B computes the value AUTH(K_{ab},N_{ab},N_{ba},B) based upon at least the new key K_{ab}, A's nonce N_{ab}, B's nonce N_{ba} and the identity of B. At the end of block 804, B transmits AUTH(K_{as},N_{as},K_{ab},B), BRAID(K_{as},$\hat{N}_{sa}$,K_{ab}) to A. At block 805, B transmits AUTH(K_{ab},N_{ab},N_{ba},B) and its nonce N_{ba} to A.

Referring to block 806, user A receives the above message and extracts the new key K_{ab} from the value of the user's function BRAID(K_{as},$\hat{N}_{ab}$,K_{ab}) using the key K_{as} and the value AUTH(K_{as},N_{as},K_{ab},B). Then, A authenticates the new key K_{ab} by verifying the authentication function AUTH(K_{as},N_{as},K_{ab},B) based upon at least the user's key K_{as}, the value N_{as}, the new key K_{ab}, and the identity of the other user B.

Then, A verifies the received value AUTH(K_{ab},N_{ab},N_{ba},B) based upon at least the new key K_{ab}, A's nonce N_{ab}, B's nonce N_{ba}, and the identity of B.

Still referring to block 806, user A computes the value of another authentication function ACK(K_{as},N_{as},K_{ab}) based at least on the user's key K_{as}, the user's nonce for S (N_{as}), and the new key K_{ab} in order to complete two-way authentication between A and S. User A also computes yet another authentication function ACK(K_{ab},N_{ab},N_{ba}) based upon at least the new key K_{ab}, A's nonce for B, N_{ab}, and B's nonce for A, N_{ba}. At the end of block 806, user A transmits to B ACK(K_{as},F_{as},K_{ab}) and ACK(K_{ab},N_{ab},N_{ba}).

Referring now to block 807, user B receives ACK(K_{as},N_{as},K_{ab}) and ACK(K_{ab},N_{ab},N_{ba}). First, user B recomputes the value of the function ACK(K_{ab},N_{ab},N_{ba}) and compares it with its counterpart in the messages received from A; a

match results in the authentication of A.

User B now computes the value of another authentication function ACK(K_{bs},N_{bs},K_{ab}) based at least on the user's key K_{bs}, the user's nonce for S (N_{bs}), and the new key K_{ab} in order to complete two-way authentication between B and S. Then, user B transmits to S ACK(K_{as},N_{as},K_{ab}) and ACK(K_{bs},N_{bs},K_{ab}).

Referring still to block 807, server S receives ACK(K_{as},N_{as},K_{ab}) and ACK(K_{bs},N_{bs},K_{ab}) sent by B. Server S first recomputes the value of the authentication function ACK(K_{as},N_{as},K_{ab}) and compares it with its counterpart in the message received from B; a match results in the authentication of A. Next, server S first recomputes the value of the authentication function ACK(K_{bs},N_{bs},K_{ab}) and compares it with its counterpart in the message received from B; a match results in the authentication of B.

This completes the protocol.

Claims

1. A method for providing secure, authenticated distribution in a communication system of a key to be used by a group of selected users, the key being distributed from a server to a sub-group consisting of the users of said group other than said server, each user of said sub-group sharing a secret user key with said server, said method comprising the steps of:

   - transmitting (501,511) the user's identification (A), a user's nonce (N_{as}) and an identification (B) of a user of said sub-group other than said transmitting user from each user of said sub-group to said server through an available path of said system;
   - generating (502) a new, common group key (K_{ab}) for all users of said group by said server;
   - computing (503,512) the value of a first function for each user of said sub-group by said server, said first function depending upon at least said user's key (K_{as}), said user's nonce (N_{as}), said group key (K_{ab}) and the identification (B) of the user of said group other than said user;
   - computing (504,513) the value of a second function for each user of said sub-group by said server, said second function depending upon at least said user's key (K_{as}), the value of said user's first function and said group key (K_{ab});
   - transmitting (505,514) the values of said user's first and second functions from said server to each user of said sub-group through an available path of said system;
   - extracting (506,515) said group key, by each user of said sub-group, from the value of said user's second function, employing said user's key and the value of said user's first function; and
   - re-computing (507,516) the value of said user's first function by each user of said sub-group and considering authenticated said extracted group key (K_{ab}) if said re-computed value is equal to the value of said user's first function received by said server.

2. The method according to claim 1, further comprising the steps of:

   - computing (508,517) the value of a third function by each user of said sub-group, said user's third function depending upon at least said user's key (K_{as}), said user's nonce (N_{as}) and said group key (K_{ab});
   - transmitting (509,518) the value of said user's third function from each user of said sub-group to said server through an available path of said system; and
   - re-computing (510,519) the value of said user's third function for each user of said sub-group by said server and authenticating each user of said sub-group if said re-computed value if equal to the value of said user's third function received by said user.

3. The method according to any previous claim, further comprising the step of:

   - each user of said sub-group authenticating (520 to 526) itself to any other user of said sub-group by encrypting and exchanging information with said group key (K_{ab}) between said users.

4. The method according to any previous claim, wherein said first, second and third functions are encryption functions under said user's key (K_{as}).

5. The method according to any previous claim, wherein said second function is an encryption function under said user's key (K_{as}) of at least said user's first function, exclusively OR-ed with said group key (K_{ab}).

6. The method according to claim 5, wherein said group key (K_{ab}) is extracted by each user of said sub-group by re-computing the value of said user's encryption function and XOR-ing said result with the value of said user's second function.

7. The method according to any previous claim, further characterized in that:

- the information to be transmitted from each user of said sub-group to the server is collected by one of the users of the system and then transmitted from said one user to said server.

8. The method according to any previous claim, further characterized in that:

- the information to be transmitted over the system is identified by a tag identifying the addressed user or server.

9. A communication system (100) for providing secure, authenticated distribution of a key to be used by a group of selected users (101,102,103,109), the key being distributed from a server (109) to a sub-group of users (101,102,103) consisting of the users of said group other than said server, each user of said sub-group sharing a corresponding secret user key with said server, said system comprising:

- means for transmitting (501,511) the user's identification, a user's nonce and an identification of a user of said sub-group other than said user from each user of said sub-group to said server through an available path of said system;
- means for generating (502) a new, common group key for all users of said group by said server;
- means for computing (503,512) the value of a first function for each user of said sub-group by said server, said first function depending upon at least said user's key, said user's nonce, said group key and the identification of the user of said group other than said user;
- means for computing (504,513) the value of a second function for each user of said sub-group by said server, said second function depending upon at least said user's key, the value of said user's first function and said group key;
- means for transmitting (505,514) the values of said user's first and second functions from said server to each user of said sub-group through an available path of said system;
- means for extracting (506,515) said group key, by each user of said sub-group, from the value of said user's second function, employing said user's key and the value of said user's first function; and
- means for re-computing (507,516) the value of said user's first function by each user of said sub-group and considering authenticated said extracted group key if said re-computed value is equal to the value of said user's first function received by said server.

10. The communication system according to claim 9, further comprising:

- means for computing (508,517) the value of a third function by each user of said sub-group, said user's third function depending upon at least said user's key, said user's nonce and said group key;
- means for transmitting (509,518) the value of said user's third function from each user of said sub-group to said server through an available path of said system; and
- means for re-computing (510,519) the value of said user's third function for each user of said sub-group by said server and authenticating each user of said sub-group if said re-computed value if equal to the value of said user's third function received by said user.

Patentansprüche

1. Ein Verfahren zu sicheren, authentifizierten Verteilung eines Schlüssels in einem Kommunikationssystem, der von einer Gruppe ausgewählter Benutzer verwendet werden soll, wobei der Schlüssel von einem Server an eine Teilgruppe verteilt wird, die aus den Benutzern der genannten Gruppe außer dem genannten Server besteht, wobei jeder Benutzer der genannten Teilgruppe mit dem genannten Server einen geheimen Benutzerschlüssel gemeinsam nutzt und wobei das genannte Verfahren die folgenden Schritte umfaßt:

- Übertragen (501, 511) der Benutzerkennung (A), einer Gelegenheitsbildung (N_{as}) des Benutzers und einer Kennung (B) eines Benutzers der genannten Teilgruppe, der nicht der genannte übertragende Benutzer ist, von jedem Benutzer der genannten Teilgruppe zu dem genannten Server über einen verfügbaren Pfad des

genannten Systems;

- Erzeugen (502) eines neuen gemeinsamen Gruppenschlüssels (K_{ab}) für alle Benutzer der genannten Gruppe durch den genannten Server;
- Berechnen (503, 512) des Werts einer ersten Funktion für jeden Benutzer der genannten Teilgruppe durch den genannten Server, wobei die genannte erste Funktion mindestens abhängt von dem Schlüssel des genannten Benutzers (K_{as}), der Gelegenheitsbildung des genannten Benutzers (N_{as}), dem genannten Gruppenschlüssel (K_{ab}) und der Kennung (B) des Benutzers der genannten Gruppe, außer dem genannten Benutzer;
- Berechnen (504, 513) des Werts einer zweiten Funktion für jeden Benutzer der genannten Teilgruppe durch den genannten Server, wobei die genannte zweite Funktion mindestens von dem Schlüssel des genannten Benutzers (K_{as}), dem Wert der ersten Funktion des genannten Benutzers und dem genannten Gruppenschlüssel (K_{ab}) abhängt;
- Übertragen (505, 514) der Werte der ersten und der zweiten Funktion des genannten Benutzers von dem genannten Server an jeden Benutzer der genannten Teilgruppe über einen verfügbaren Pfad des genannten Systems;
- Extrahieren (506, 515) des genannten Gruppenschlüssels durch jeden Benutzer der genannten Teilgruppe aus dem Wert der zweiten Funktion des genannten Benutzers, unter Anwendung des Schlüssels des genannten Benutzers und des Werts der ersten Funktion des genannten Benutzers; und
- Neuberechnen (507, 516) des Werts der ersten Funktion des genannten Benutzers durch jeden Benutzer der genannten Teilgruppe und Betrachten des genannten extrahierten Gruppenschlüssels (K_{ab}) als authentifiziert, wenn der genannte neuberechnete Wert dem Wert der ersten Funktion des genannten Benutzers entspricht, die von dem genannten Server empfangen wurde.

2. Das Verfahren nach Anspruch 1, weiter folgende Schritte umfassend:

- Berechnen (508, 517) des Werts einer dritten Funktion durch jeden Benutzer der genannten Teilgruppe, wobei die dritte Funktion des genannten Benutzers mindestens abhängt vom Schlüssel (K_{as}) des genannten Benutzers, der Gelegenheitsbildung (N_{as}) des genannten Benutzers und dem genannten Gruppenschlüssel (K_{ab});
- Übertragen (509, 518) des Werts der dritten Funktion des genannten Benutzers von jedem Benutzer der genannten Teilgruppe zum genannten Server über einen verfügbaren Pfad des genannten Systems; und
- Neuberechnen (510, 519) des Werts der dritten Funktion des genannten Benutzers für jeden Benutzer der genannten Teilgruppe durch den genannten Server und Authentifizieren jedes Benutzers der genannten Teilgruppe, wenn der genannte neuberechnete Wert dem Wert der von dem genannten Benutzer empfangenen dritten Funktion des genannten Benutzers entspricht.

3. Das Verfahren nach jedem vorangehenden Anspruch, weiter folgenden Schritt umfassend:

- Jeder Benutzer der genannten Teilgruppe authentifiziert sich selbst (520 bis 526) gegenüber jedem anderen Benutzer der genannten Teilgruppe durch Verschlüsselung und Austauschen von Informationen mit dem genannten Gruppenschlüssel (K_{ab}) unter den genannten Benutzern.

4. Das Verfahren nach jedem vorangehenden Anspruch, bei dem die genannte erste, zweite und dritte Funktion Verschlüsselungsfunktionen unter dem Schlüssel des genannten Benutzers (K_{as}) sind.

5. Das Verfahren nach jedem vorangehenden Anspruch, bei dem die genannte zweite Funktion eine Verschlüsselungsfunktion unter dem Schlüssel (K_{as}) des genannten Benutzers von mindestens der ersten Funktion des genannten Benutzers ist, die mit dem genannten Gruppenschlüssel (K_{ab}) exklusiv geODERt wurde.

6. Das Verfahren nach Anspruch 5, bei dem der genannte Gruppenschlüssel (K_{ab}) von jedem Benutzer der genannten Teilgruppe extrahiert wird durch Neuberechnen des Werts der Verschlüsselungsfunktion des genannten Benutzers und durch XODERn des genannten Ergebnisses mit dem Wert der zweiten Funktion des genannten Benutzers.

7. Das Verfahren nach jedem vorangehenden Anspruch, weiter dadurch gekennzeichnet, daß:

- die von jedem Benutzer der genannten Teilgruppe an den Server zu übertragende Information von einem Benutzer des Systems gesammelt und dann von dem genannten einen Benutzer an den genannten Server übertragen wird.

8. Das Verfahren nach jedem vorangehenden Anspruch, weiter dadurch gekennzeichnet, daß:

- die über das System zu übertragende Information mit einem Identifizierungskennzeichen versehen wird, welches den adressierten Benutzer oder Server identifiziert.

9. Ein Kommunikationssystem (100) zur sicheren und authentifizierten Verteilung eines von einer Gruppe ausgewählter Benutzer (101, 102, 103, 109) zu benutzenden Schlüssels, wobei der Schlüssel von einem Server (109) an eine Teilgruppe von Benutzern (101, 102, 103) verteilt wird, die aus den Benutzern der genannten Gruppe, außer dem genannten Server, besteht, wobei jeder Benutzer der genannten Teilgruppe einen entsprechenden geheimen Benutzerschlüssel gemeinsam mit dem genannten Server nutzt und wobei das genannte System folgendes umfaßt:

- Mittel zum Übertragen (501, 511) der Benutzerkennung, einer Gelegenheitsbildung des Benutzers und einer Kennung eines Benutzers der genannten Teilgruppe, der nicht der genannte Benutzer ist, von jedem Benutzer der genannten Teilgruppe an den genannten Server über einen verfügbaren Pfad des genannten Systems;
- Mittel zur Erzeugung (502) eines neuen gemeinsamen Gruppenschlüssels für alle Benutzer der genannten Gruppe durch den genannten Server;
- Mittel zur Berechnung (503, 512) des Werts einer ersten Funktion für jeden Benutzer der genannten Teilgruppe durch den genannten Server, wobei die genannte erste Funktion mindestens abhängt von dem Schlüssel des genannten Benutzers, der Gelegenheitsbildung des genannten Benutzers, dem genannten Gruppenschlüssel und der Kennung des Benutzers der genannten Gruppe, der nicht der genannte Benutzer ist;
- Mittel zur Berechnung (504, 513) des Werts einer zweiten Funktion für jeden Benutzer der genannten Teilgruppe durch den genannten Server, wobei die genannte zweite Funktion mindestens abhängt von dem Schlüssel des genannten Benutzers, dem Wert der ersten Funktion des genannten Benutzers und dem genannten Gruppenschlüssel;
- Mittel zur Übertragung (505, 514) der Werte der ersten und der zweiten Funktion des genannten Benutzers von dem genannten Server an jeden Benutzer der genannten Teilgruppe über einen verfügbaren Pfad des genannten Systems;
- Mittel zum Extrahieren (506, 515) des genannten Gruppenschlüssels durch jeden Benutzer der genannten Teilgruppe aus dem Wert der zweiten Funktion des genannten Benutzers, unter Anwendung des Schlüssels des genannten Benutzers und des Werts der ersten Funktion des genannten Benutzers; und
- Mittel zur Neuberechnung (507, 516) des Werts der ersten Funktion des genannten Benutzers durch jeden Benutzer der genannten Teilgruppe und Betrachten des genannten extrahierten Gruppenschlüssels als authentifiziert, wenn der genannte neuberechnete Wert dem Wert der ersten Funktion des genannten Benutzers entspricht, die von dem genannten Server empfangen wurde.

10. Das Kommunikationssystem nach Anspruch 9, weiter folgendes umfassend:

- Mittel zur Berechnung (508, 517) des Werts einer dritten Funktion durch jeden Benutzer der genannten Teilgruppe, wobei die dritte Funktion des genannten Benutzers mindestens abhängt von dem Schlüssel des genannten Benutzers, der Gelegenheitsbildung des genannten Benutzers und dem genannten Gruppenschlüssel;

- Mittel zur Übertragung (509, 518) des Werts der dritten Funktion des genannten Benutzers von jedem Benutzer der genannten Teilgruppe an den genannten Server über einen verfügbaren Pfad des genannten Systems; und

- Mittel zur Neuberechnung (510, 519) des Werts der dritten Funktion des genannten Benutzers für jeden Benutzer der genannten Teilgruppe durch den genannten Server und Authentifizierung jedes Benutzers der genannten Teilgruppe, wenn der genannte neuberechnete Wert dem Wert der dritten Funktion des genannten Benutzers entspricht, der von dem genannten Benutzer empfangen wurde.

Revendications

1. Un procédé pour fournir une distribution sécurisée, authentifiée, dans un système de communication, d'une clé devant être utilisée par un groupe d'utilisateurs sélectionnés, la clé étant distribuée à partir d'un serveur à un sous-groupe composé des utilisateurs dudit groupe autre que ledit serveur, chaque utilisateur dudit groupe partageant une clé d'utilisateur secrète avec ledit serveur, ledit procédé comprenant les étapes de :

- transmission (501, 511) de l'identification de l'utilisateur (A), d'un mot de passe de l'utilisateur (N_{as}) et d'une

identification (B) d'un utilisateur dudit sous-groupe autre que ledit utilisateur transmettant venant de chaque utilisateur dudit sous-groupe audit serveur, au travers d'un chemin disponible dudit système;

- création (502) d'une nouvelle clé (K_{ab}) commune au groupe pour tous les utilisateurs dudit groupe par ledit serveur;

- calcul (503, 512) de la valeur d'une première fonction pour chaque utilisateur dudit sous-groupe par ledit serveur, ladite première fonction dépendant au moins de ladite clé d'utilisateur (K_{as}), dudit mot de passe de l'utilisateur (N_{as}), de ladite clé de groupe (K_{ab}) et de ladite identification (B) de l'utilisateur dudit groupe autre que ledit utilisateur;

- calcul (504, 513) de la valeur d'une deuxième fonction pour chaque utilisateur dudit sous-groupe par ledit serveur, ladite deuxième fonction dépendant au moins de ladite clé d'utilisateur (K_{as}), de la valeur de ladite première fonction et de ladite clé de groupe (K_{ab});

- transmission (505, 514) des valeurs desdites première et deuxième fonctions à partir dudit serveur vers chaque utilisateur dudit sous-groupe au travers d'un chemin disponible dudit système;

- extraction (506, 515) de ladite clé de groupe, par chaque utilisateur dudit sous-groupe, à partir de la valeur de ladite deuxième fonction de l'utilisateur, en utilisant ladite clé d'utilisateur et la valeur de ladite première fonction de l(utilisateur; et

- re-calcul (507, 516) de la valeur de ladite première fonction de l'utilisateur par chaque utilisateur dudit sous-groupe et considération de ladite clé de groupe extraite (K_{ab}) comme étant authentifiée, si ladite valeur recalculée est égale à la valeur de ladite première fonction de l'utilisateur reçue par ledit serveur.

2. Le procédé selon la revendication 1, comprenant en outre les étapes de :

- calcul (508, 517) de la valeur d'une troisième fonction par chaque utilisateur dudit sous-groupe, ladite troisième fonction de l'utilisateur dépendant au moins de ladite clé d'utilisateur (K_{as}), dudit mot de passe de l'utilisateur (N_{as}) et de ladite clé de groupe (K_{ab});

- transmission (509, 518) de la valeur de ladite troisième fonction de l'utilisateur venant de chaque utilisateur dudit sous-groupe vers ledit serveur au travers d'un chemin disponible dudit système; et

- re-calcul (510, 519) de la valeur de ladite troisième fonction de l'utilisateur pour chaque utilisateur dudit sous-groupe par ledit serveur et authentification de chaque utilisateur dudit sous-groupe par ledit serveur, si ladite valeur recalculée est égale à la valeur de ladite troisième fonction de l'utilisateur reçue par ledit utilisateur.

3. Le procédé selon l'une quelconque des revendications précédentes, comprenant en outre les étapes de :

- authentification (520 à 526) par lui même de chaque utilisateur dudit sous-groupe, de tout autre utilisateur dudit sous-groupe en codant et en échangeant une information avec ladite clé de groupe (K_{ab}) entre lesdits utilisateurs.

4. Le procédé selon l'une quelconque des revendications précédentes, dans lequel lesdites première, deuxième et troisième fonctions sont des fonctions de codage avec ladite clé d'utilisateur (K_{as}).

5. Le procédé selon l'une quelconque des revendications précédentes, dans lequel ladite deuxième fonction est une fonction de codage avec ladite clé d'utilisateur (K_{as}) d'au moins ladite première fonction, soumise à une fonction OU exclusif avec ladite clé de groupe (K_{ab}).

6. Le procédé selon la revendication 5, dans lequel ladite clé de groupe (K_{ab}) est extraite par chaque utilisateur dudit sous-groupe en recalculant la valeur de ladite fonction de codage de l'utilisateur et en soumettant à une fonction OU exclusif ledit résultat avec la valeur de ladite deuxième fonction de l'utilisateur.

7. Le procédé selon l'une quelconque des revendications précédentes, caractérisé, en outre, en ce que :

- l'information devant être transmise par chaque utilisateur dudit sous-groupe audit serveur est collectée par un des utilisateurs du système, puis transmise du dit utilisateur audit serveur.

8. Le procédé selon l'une quelconque des revendications précédentes, caractérisé, en outre, en ce que :

- l'information devant être transmise dans le système est identifiée par une étiquette identifiant l'utilisateur ou le serveur auquel elle est adressée.

9. Un système de communication (100) pour fournir une distribution sécurisée, authentifiée d'une clé devant être utilisée par un groupe d'utilisateurs sélectionnés (101, 102, 103, 104), la clé étant distribuée à partir d'un serveur (109) à un sous-groupe d'utilisateurs (101, 102, 103) composé d'utilisateurs dudit groupe autre que dudit serveur, chaque utilisateur dudit groupe partageant une clé d'utilisateur secrète avec ledit serveur, ledit système comprenant :

- des moyens de transmission (501, 511) de l'identification de l'utilisateur, d'un mot de passe de l'utilisateur et d'une identification d'un utilisateur dudit sous-groupe autre que ledit utilisateur, venant de chaque utilisateur dudit sous-groupe vers ledit serveur, au travers d'un chemin disponible dudit système;

- des moyens de création (502) d'une nouvelle clé commune au groupe pour tous les utilisateurs dudit groupe par ledit serveur;

- des moyens de calcul (503, 512) de la valeur d'une première fonction pour chaque utilisateur dudit sous-groupe par ledit serveur, ladite première fonction dépendant au moins de ladite clé d'utilisateur, dudit mot de passe de l'utilisateur, de ladite clé de groupe et de l'identification de l'utilisateur dudit groupe autre que dudit utilisateur;

- des moyens de calcul (504, 513) de la valeur d'une deuxième fonction pour chaque utilisateur dudit sous-groupe par ledit serveur, ladite deuxième fonction dépendant au moins de ladite clé d'utilisateur, de la valeur de ladite première fonction et de ladite clé de groupe;

- des moyens de transmission (505, 514) des valeurs desdites première et deuxième fonctions à partir dudit serveur vers chaque utilisateur dudit sous-groupe, au travers d'un chemin disponible dudit système;

- des moyens d'extraction (506, 515) de ladite clé de groupe, par chaque utilisateur dudit sous-groupe, dans la valeur de ladite deuxième fonction de l'utilisateur, en utilisant ladite clé d'utilisateur et la valeur de ladite première fonction de l'utilisateur; et

- des moyens de re-calcul (507, 516) de la valeur de ladite première fonction de l'utilisateur par chaque utilisateur dudit sous-groupe, et des moyens de considération de ladite clé de groupe extraite comme étant authentifiée, si ladite valeur recalculée est égale à la valeur de ladite première fonction de l'utilisateur reçue par ledit serveur.

10. Le système de communication selon la revendication 9, comprenant en outre :

- des moyens de calcul (508, 517) de la valeur d'une troisième fonction par chaque utilisateur dudit sous-groupe, ladite troisième fonction de l'utilisateur dépendant au moins de ladite clé d'utilisateur, dudit mot de passe de l'utilisateur et de ladite clé de groupe;

- des moyens de transmission (509, 518) de la valeur de ladite troisième fonction de l'utilisateur venant de chaque utilisateur dudit sous-groupe vers ledit serveur au travers d'un chemin disponible dudit système; et

- des moyens de re-calcul (510, 519) de la valeur de ladite troisième fonction de l'utilisateur pour chaque utilisateur dudit sous-groupe, par ledit serveur et une authentification de chaque utilisateur dudit sous-groupe par ledit serveur, si ladite valeur recalculée est égale à la valeur de ladite troisième fonction de l'utilisateur reçue par ledit utilisateur.

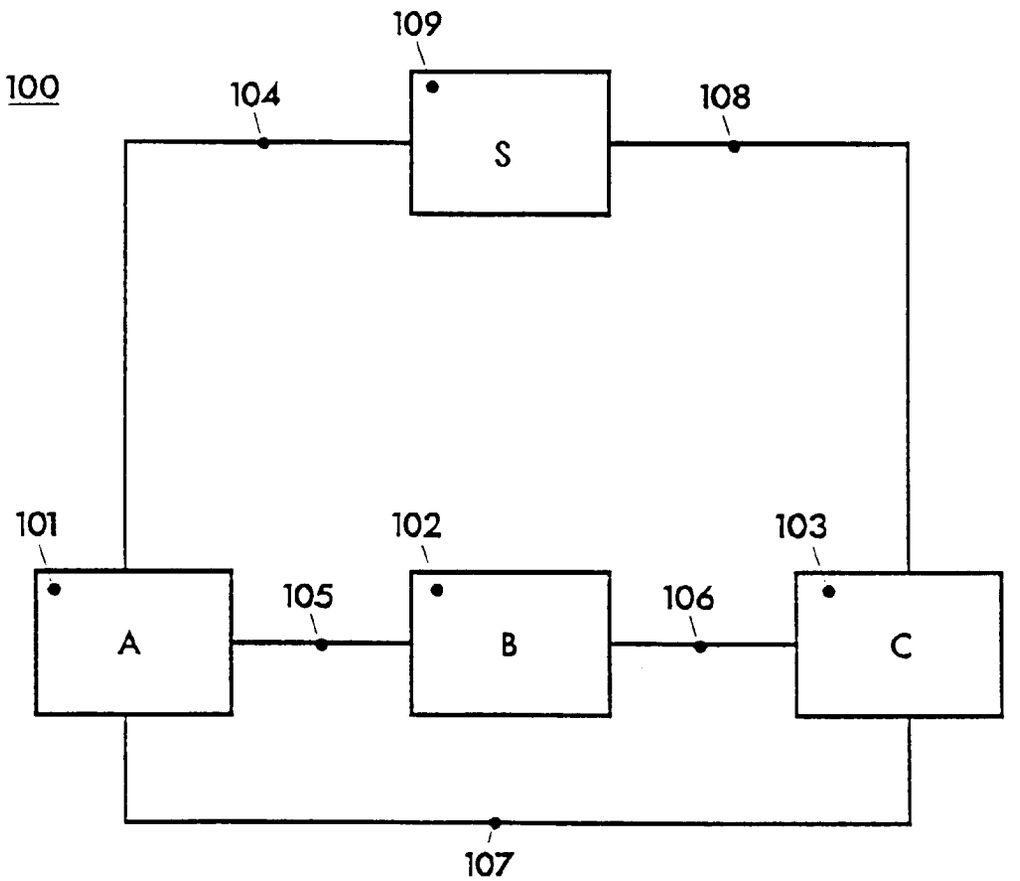

Fig. 1

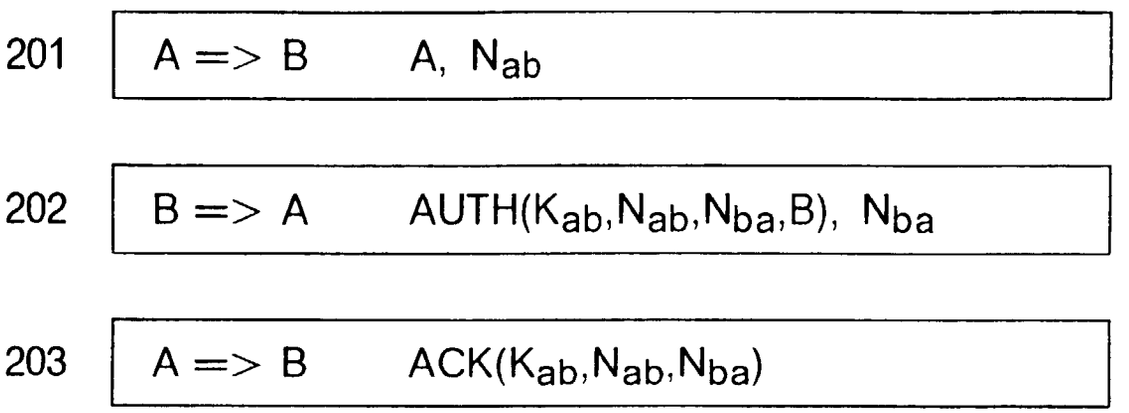

Fig. 2

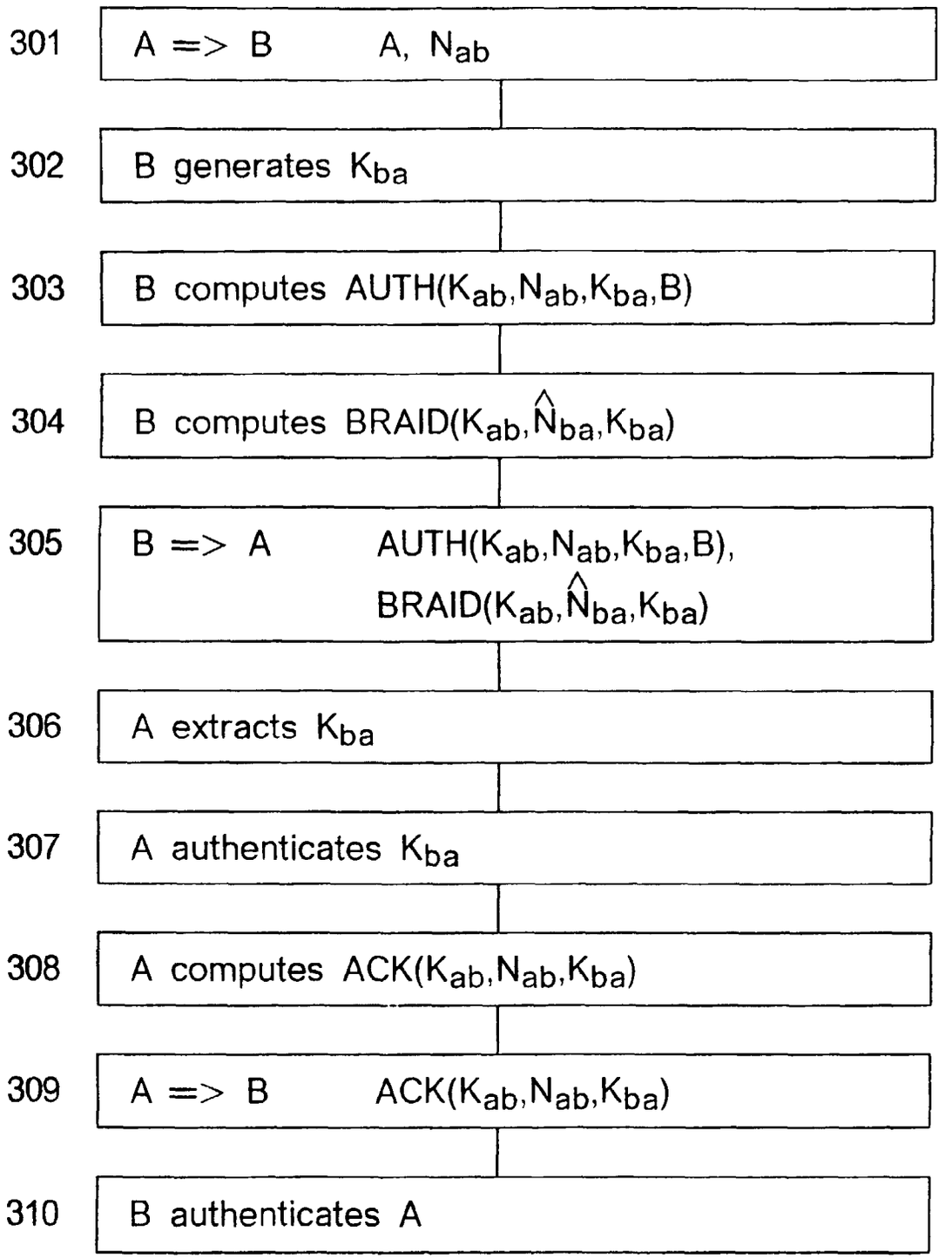
301
A => B A, N_{ab}
302
B generates K_{ba}
303
B computes AUTH(K_{ab},N_{ab},K_{ba},B)
304
B computes BRAID(K_{ab},$\hat{N}_{ba}$,K_{ba})
305
B => A AUTH(K_{ab},N_{ab},K_{ba},B),
BRAID(K_{ab},$\hat{N}_{ba}$,K_{ba})
306
A extracts K_{ba}
307
A authenticates K_{ba}
308
A computes ACK(K_{ab},N_{ab},K_{ba})
309
A => B ACK(K_{ab},N_{ab},K_{ba})
310
B authenticates A

Fig. 3

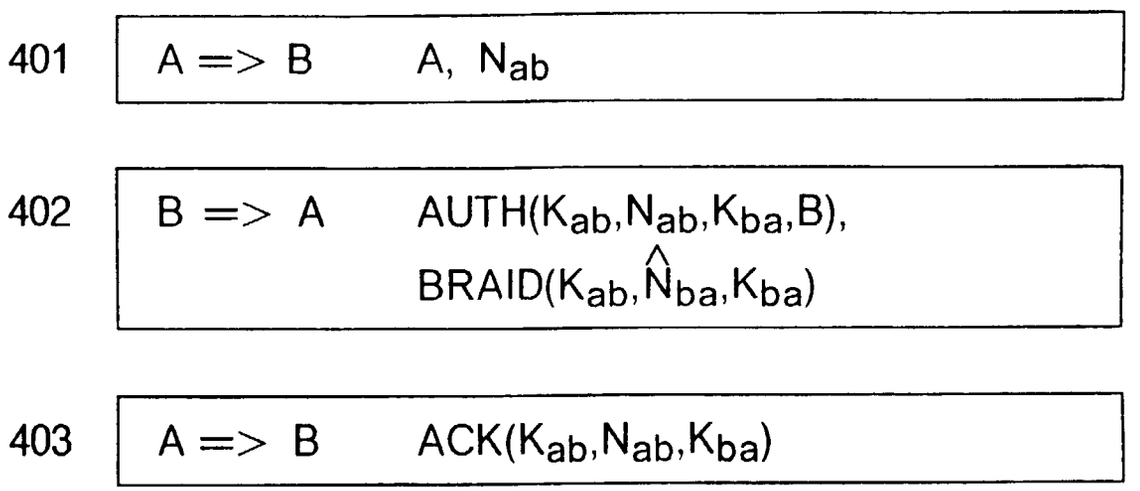
401
A => B
A, N_{ab}
402
B => A
AUTH(K_{ab},N_{ab},K_{ba},B),
BRAID(K_{ab},$\hat{N}_{ba}$,K_{ba})
403
A => B
ACK(K_{ab},N_{ab},K_{ba})

Fig. 4

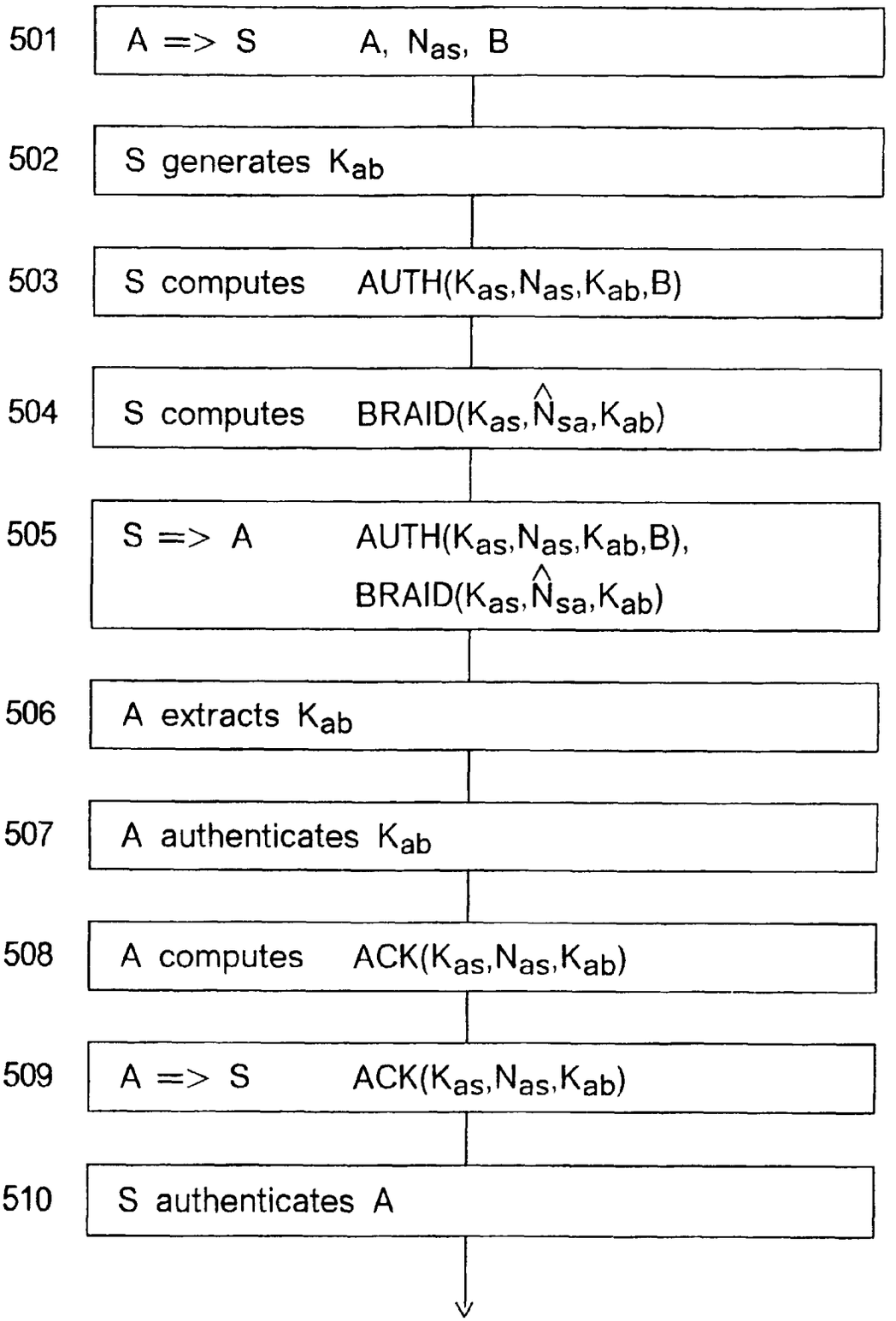
501
A => S A, N_{as}, B
502
S generates K_{ab}
503
S computes AUTH(K_{as},N_{as},K_{ab},B)
504
S computes BRAID(K_{as},$\hat{N}_{sa}$,K_{ab})
505
S => A AUTH(K_{as},N_{as},K_{ab},B),
BRAID(K_{as},$\hat{N}_{sa}$,K_{ab})
506
A extracts K_{ab}
507
A authenticates K_{ab}
508
A computes ACK(K_{as},N_{as},K_{ab})
509
A => S ACK(K_{as},N_{as},K_{ab})
510
S authenticates A

Fig. 5b

Fig. 5a

Fig. 5a

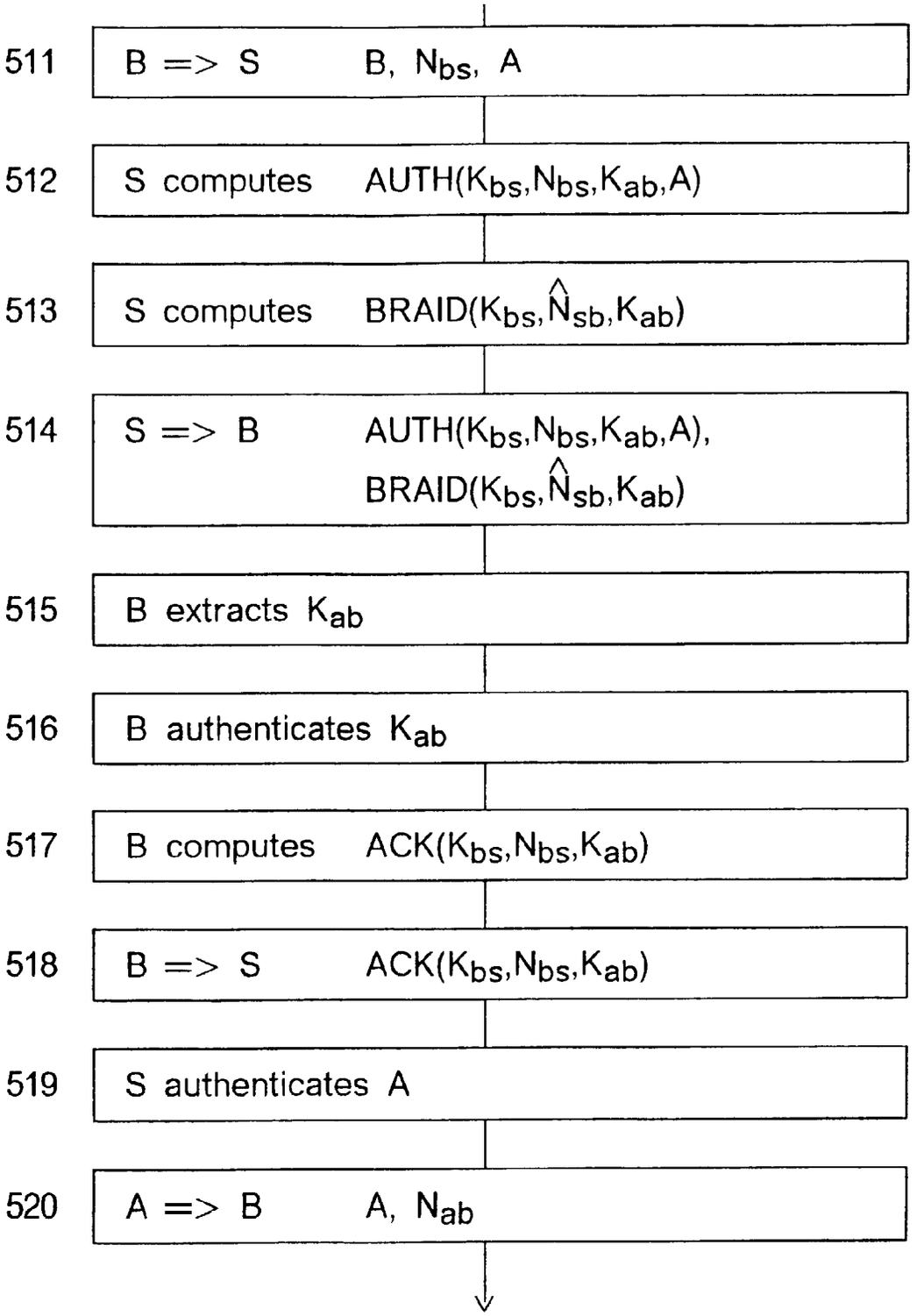

Fig. 5c

Fig. 5b

Fig. 5b

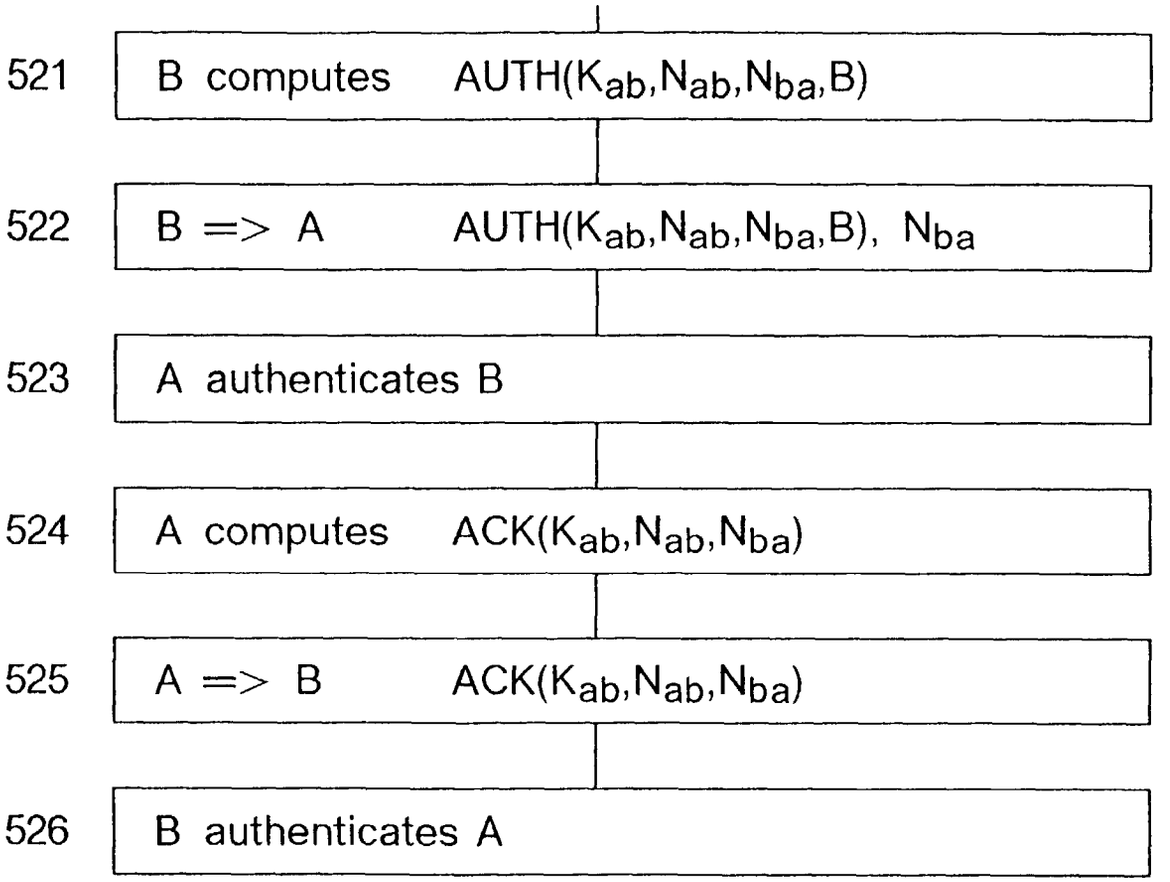
521
B computes AUTH(K_{ab},N_{ab},N_{ba},B)
522
B => A AUTH(K_{ab},N_{ab},N_{ba},B), N_{ba}
523
A authenticates B
524
A computes ACK(K_{ab},N_{ab},N_{ba})
525
A => B ACK(K_{ab},N_{ab},N_{ba})
526
B authenticates A

Fig. 5c

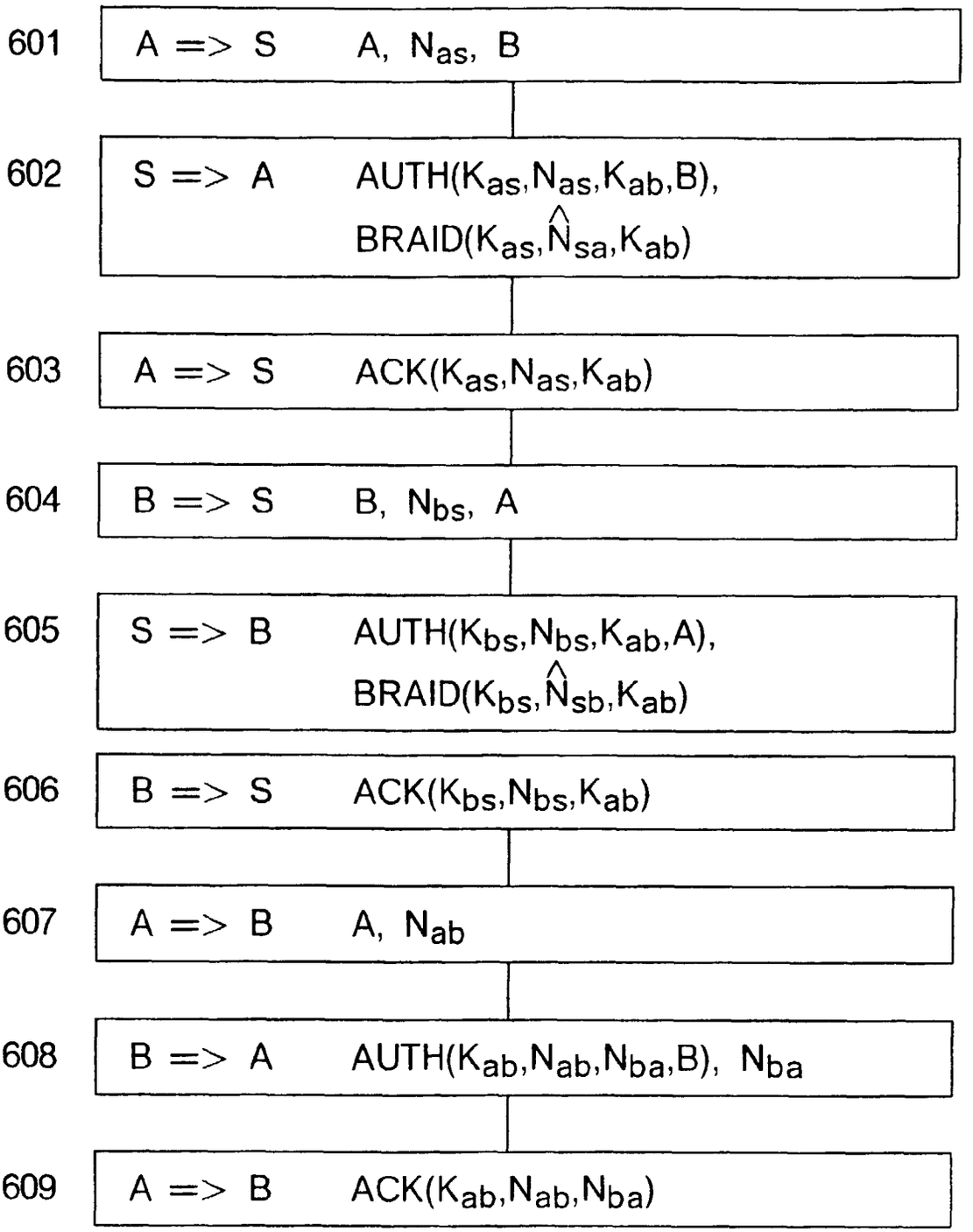
601
A => S A, N_{as}, B
602
S => A AUTH(K_{as},N_{as},K_{ab},B),
BRAID(K_{as},$\hat{N}_{sa}$,K_{ab})
603
A => S ACK(K_{as},N_{as},K_{ab})
604
B => S B, N_{bs}, A
605
S => B AUTH(K_{bs},N_{bs},K_{ab},A),
BRAID(K_{bs},$\hat{N}_{sb}$,K_{ab})
606
B => S ACK(K_{bs},N_{bs},K_{ab})
607
A => B A, N_{ab}
608
B => A AUTH(K_{ab},N_{ab},N_{ba},B), N_{ba}
609
A => B ACK(K_{ab},N_{ab},N_{ba})

Fig. 6

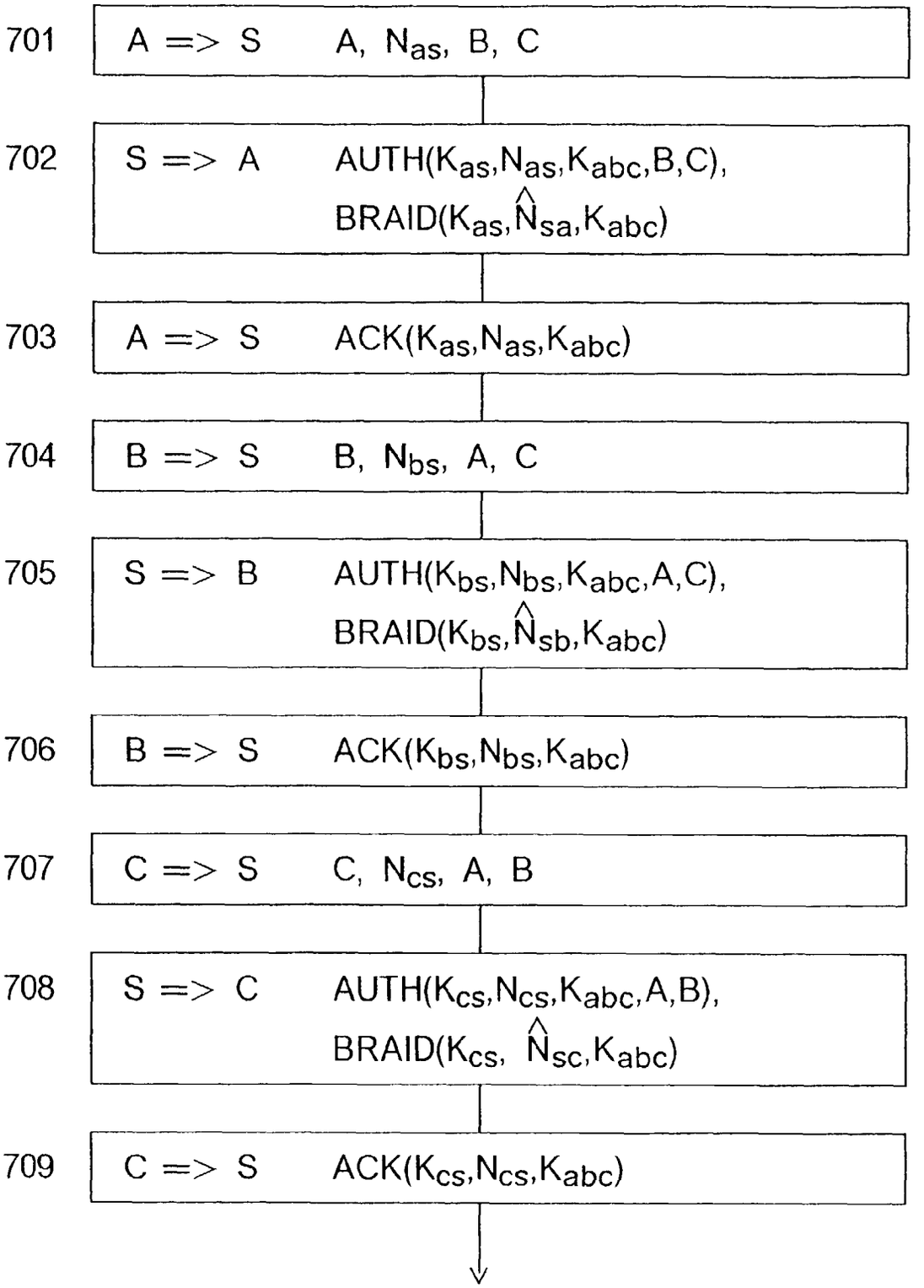
701
A => S A, N_{as}, B, C
702
S => A AUTH(K_{as},N_{as},K_{abc},B,C),
BRAID(K_{as},$\hat{N}_{sa}$,K_{abc})
703
A => S ACK(K_{as},N_{as},K_{abc})
704
B => S B, N_{bs}, A, C
705
S => B AUTH(K_{bs},N_{bs},K_{abc},A,C),
BRAID(K_{bs},$\hat{N}_{sb}$,K_{abc})
706
B => S ACK(K_{bs},N_{bs},K_{abc})
707
C => S C, N_{cs}, A, B
708
S => C AUTH(K_{cs},N_{cs},K_{abc},A,B),
BRAID(K_{cs}, $\hat{N}_{sc}$,K_{abc})
709
C => S ACK(K_{cs},N_{cs},K_{abc})

Fig. 7b

Fig. 7a

Fig. 7a

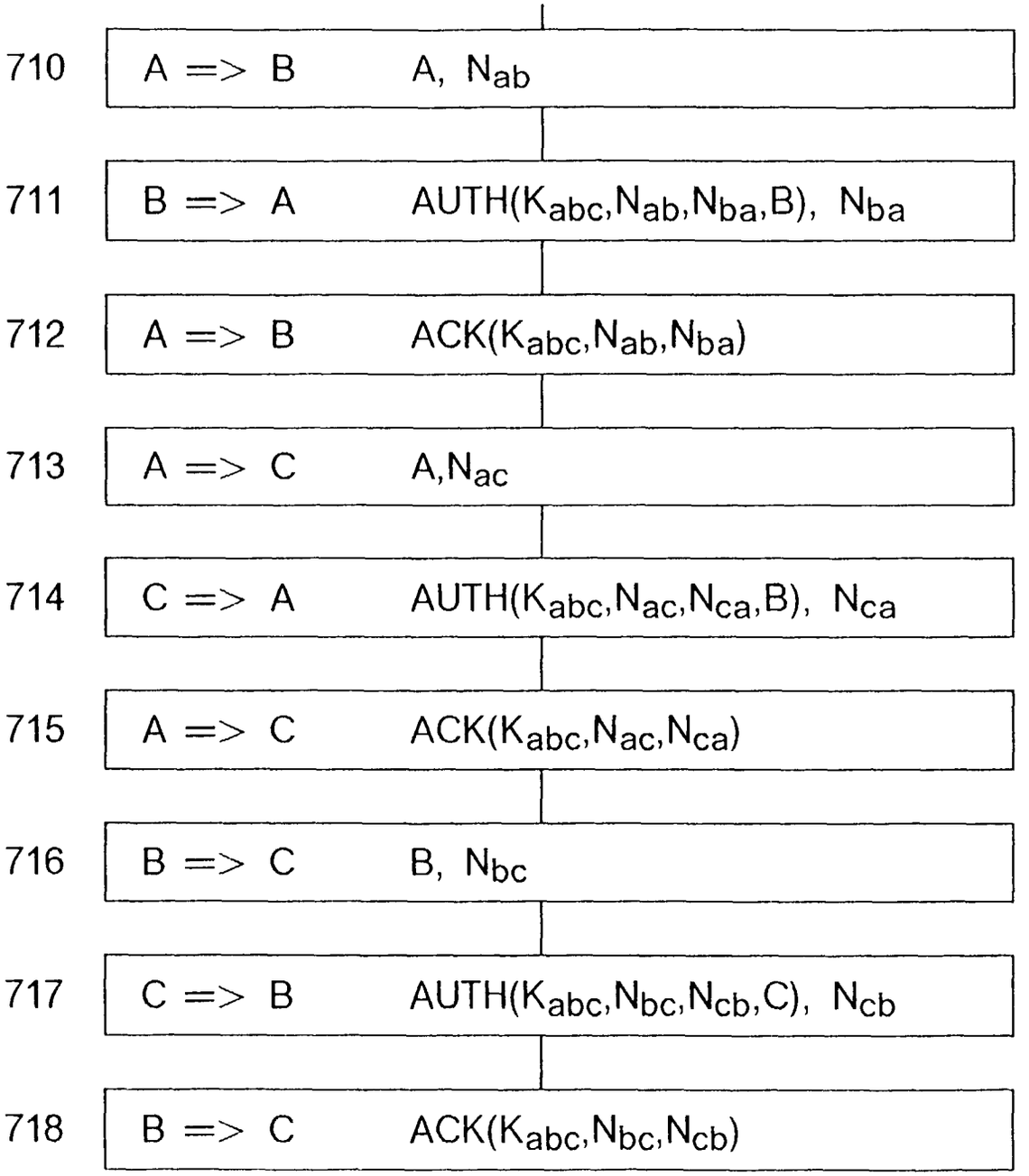
710
A => B
A, N_{ab}
711
B => A
AUTH(K_{abc},N_{ab},N_{ba},B), N_{ba}
712
A => B
ACK(K_{abc},N_{ab},N_{ba})
713
A => C
A,N_{ac}
714
C => A
AUTH(K_{abc},N_{ac},N_{ca},B), N_{ca}
715
A => C
ACK(K_{abc},N_{ac},N_{ca})
716
B => C
B, N_{bc}
717
C => B
AUTH(K_{abc},N_{bc},N_{cb},C), N_{cb}
718
B => C
ACK(K_{abc},N_{bc},N_{cb})

Fig. 7b

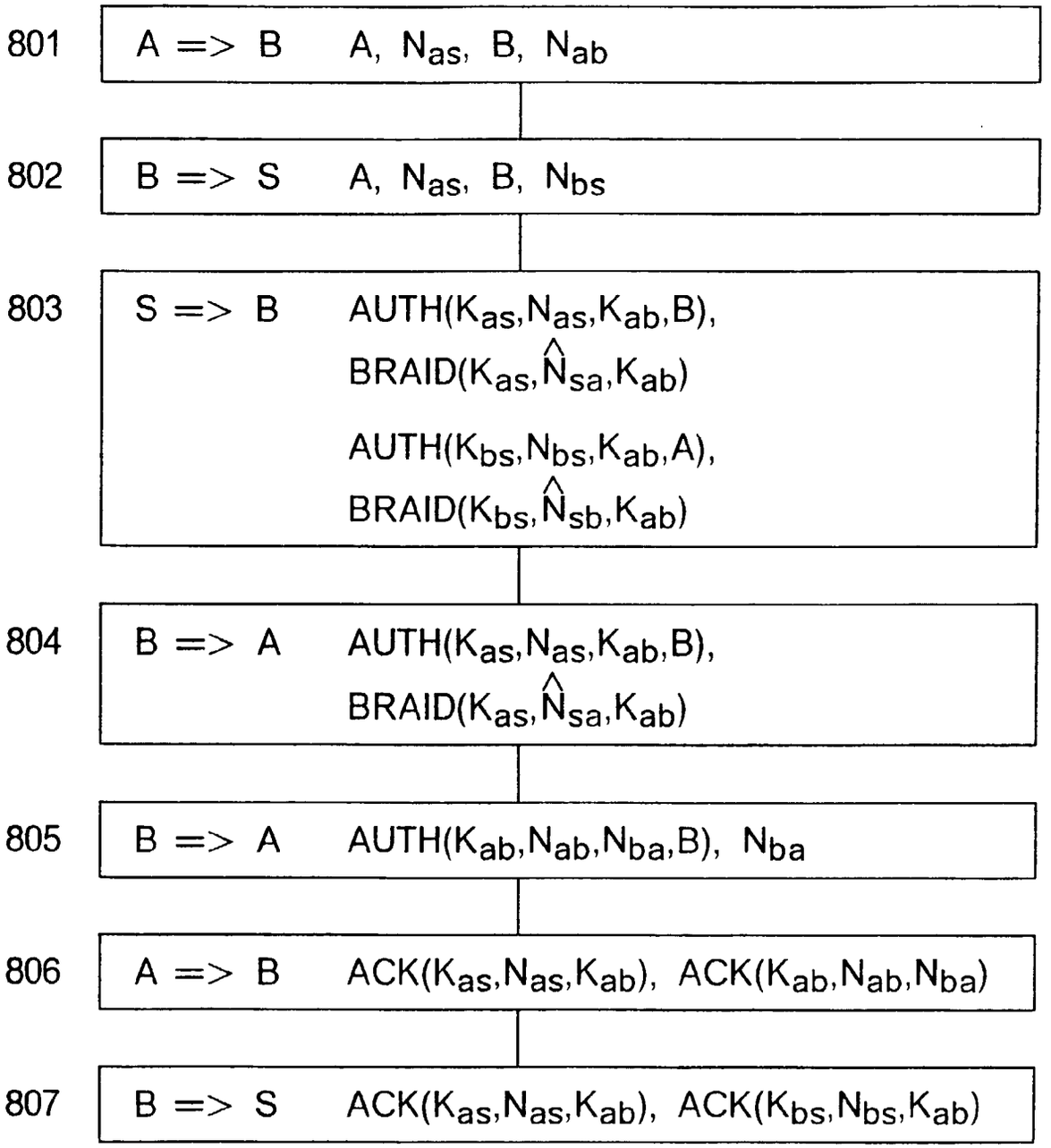
801
A => B A, N_{as}, B, N_{ab}
802
B => S A, N_{as}, B, N_{bs}
803
S => B AUTH(K_{as},N_{as},K_{ab},B),
BRAID(K_{as},$\hat{N}_{sa}$,K_{ab})
AUTH(K_{bs},N_{bs},K_{ab},A),
BRAID(K_{bs},$\hat{N}_{sb}$,K_{ab})
804
B => A AUTH(K_{as},N_{as},K_{ab},B),
BRAID(K_{as},$\hat{N}_{sa}$,K_{ab})
805
B => A AUTH(K_{ab},N_{ab},N_{ba},B), N_{ba}
806
A => B ACK(K_{as},N_{as},K_{ab}), ACK(K_{ab},N_{ab},N_{ba})
807
B => S ACK(K_{as},N_{as},K_{ab}), ACK(K_{bs},N_{bs},K_{ab})

Fig. 8